

Written Exam Corrector (WEC): Plataforma para la asistencia a la corrección de exámenes

Pol Sánchez Manzano

Resumen—WEC (Written Exam Corrector) es una herramienta web originada de la necesidad de poder coordinar y agilizar la tarea de corrección de exámenes en físico. Uno de los retos de proyecto ha sido la realización de un módulo de Reconocimiento Óptico de Caracteres (OCR) que permite al profesor identificar los alumnos sin la necesidad de hacerlo manualmente mediante el análisis automático del campo del NIA.. La aplicación web proporciona la funcionalidad de poder crear cursos, añadir nuevas pruebas a estos cursos y gestionar las correcciones de cada una de las pruebas mediante un delimitador de respuestas que permite acotar la región donde el alumno ha respondido la pregunta, de esta forma se podrá visualizar las respuestas y corregir de forma más ágil.

Palabras clave— Reconocimiento Óptico de Caracteres (OCR), Visión por Computador, Deep Learning, Aplicación Web, Corrección de Exámenes.

Abstract— WEC (Written Exam Corrector) is a web tool originated from the need to be able to coordinate and speed up the physical exam correction task. One of the project challenges has been the development of an Optical Character Recognition (OCR) module that allows the teacher to identify students without the need to do it manually by means of the automatic analysis of the NIA field.. The web application provides the functionality of being able to create courses, add new tests to these courses and manage the corrections of each one of the tests by means of an answer delimiter that allows to mark off the region where the student has answered the question, in this way you can visualize responses and correct more quickly.

Index Terms— Optical Character Recognition (OCR), Computer Vision, Deep Learning, Web application, Exam Correction.

◆

1 INTRODUCCIÓN

Una de las tareas más frecuentes y costosas para los profesores es la corrección de exámenes. Esta tarea implica gran dedicación temporal y un cuidado extremo a la hora de evaluar de manera idéntica exámenes de alumnos diferentes. Esta tarea se complica si un mismo examen ha de ser corregido por varios profesores: en este caso, además de tener que lidiar con el intercambio de documentos físicos entre personas, se ha de fijar de manera clara y unívoca los criterios que se usarán para la corrección de los mismos con el fin de evitar discrepancias en la manera de corregir.

La corrección de los exámenes es un proceso poco transparente para el alumno que, en el caso de no acudir a la revisión de examen, sólo recibe como feedback la nota final alcanzada en el mismo. Esto es debido a que los exámenes escritos no se pueden facilitar a los alumnos por razones legales, por tanto, estos sólo tienen la posibilidad de consultarlos en el despacho del profesor. Para los alumnos sería muy interesante contar con el feedback completo asociado a la corrección del examen de forma que el alumno lo pueda tener en cuenta para mejorar.

Por tanto hay una necesidad de gestionar, acelerar, flexibilizar y hacer más consistente el proceso de corrección de exámenes escritos.

En este proyecto se propone el desarrollo de una aplicación web que permita gestionar el proceso de corrección de exámenes de una asignatura. Idealmente los profesores subirán los exámenes a la aplicación y ésta los separaría por preguntas y permitiría corregir cada una de ellas a partir de unos criterios también indicados por los profesores.

-
- E-mail de contacto: pol.sanchezu@e-campus.uab.cat
 - Mención realizada: *Computación*
 - Trabajo tutorizado por: F. Xavier Roca y Jorge Bernal (*Ciencias de la Computación*)
 - Curso 2019/20

2 ESTADO DEL ARTE

Para poder dar un contexto más detallado de lo que se pretende realizar en este proyecto, se ha buscado un conjunto de herramientas en las que se ha basado la idea para tomar cuales son los puntos a mejorar o los más diferenciales.

Gradescope[1]: Es una herramienta para la corrección de documentos que permite el escaneo de exámenes para después poder delimitar las preguntas que serán posteriormente evaluadas. Esta herramienta permite gestionar las rúbricas y poder añadir anotaciones de forma opcional.

Crowdmark[2]: Esta herramienta permite realizar pruebas de forma online, añadir los exámenes escaneados del alumno o bien que el alumno suba sus ejercicios con el fin de ser evaluados por parte del profesor. También permite flexibilizar el sistema de evaluación. Asimismo, realiza estadísticas para evaluar el nivel de aprendizaje alcanzado por cada alumno.

Podríamos ver nuestra herramienta como una versión adaptada de Gradescope. El factor diferenciador de este proyecto es la integración de herramientas de Visión por Computador que permita automatizar tareas como la extracción de nombre y NIU a partir de un módulo OCR y también emplear técnicas de reconocimiento de patrones para poder identificar marcas que indican dónde empiezan y acaban el examen de cada alumno. También se plantea añadir estadísticas a nivel global del conjunto de alumnos que ha realizado el examen para poder obtener feedback sobre qué partes explicadas en clase hay que hacer más hincapié.

3 OBJETIVOS

El objetivo principal de este proyecto es el desarrollo de una aplicación web que permita corregir los exámenes de cualquier asignatura y gestionar los cursos junto a las pruebas que la componen (Fig. 1).

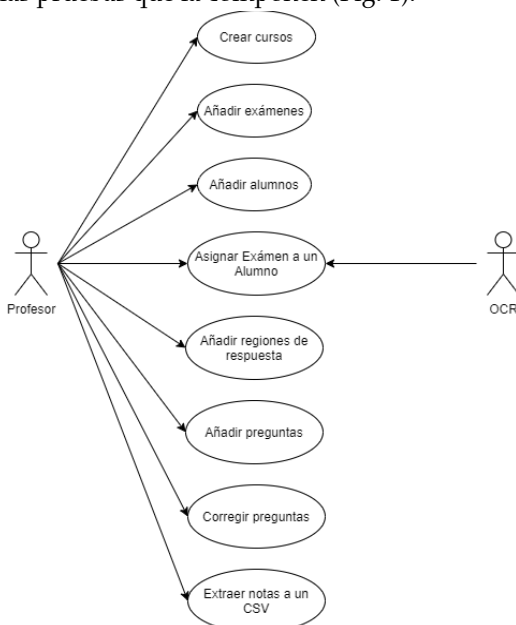


Fig. 1 Diagrama de casos de uso

Objetivos secundarios:

- Diseño de la BBDD donde se almacenará la información tanto de las asignaturas como de los exámenes a corregir
- Diseño del módulo de VC que permita extraer y reconocer el nombre y el NIU del alumno en cuestión.
- Diseño del módulo web que permita delimitar el área que ocupará la respuesta de cada alumno a cada una de las preguntas del examen.
- Diseño del módulo web que permite realizar la corrección del examen y almacenar los datos de la corrección en la BBDD.
- Diseño de una interfaz de usuario intuitiva que facilite al docente el poder realizar la corrección de la manera más eficaz posible.
- Diseño de un módulo web que permita, mediante la exploración de la BBDD, ofrecer un conjunto de estadísticas que indiquen cómo ha sido el desempeño del alumnado en el examen en concreto.

4 HERRAMIENTAS EMPLEADAS

Se resumen a continuación las herramientas que se han usado durante el proyecto:

- **Python:** Nos permitirá hacer todo el conjunto de módulos relacionados con Visión por Computador, por lo que es esencial su utilización. Se ha escogido Python frente a alternativas como C++ o Matlab por la facilidad de integración con el resto de herramientas así como por la gran cantidad de librerías y ejemplos disponibles.
- **MySQL:** El sistema ha de poder almacenar de forma estructurada los datos relacionados con las asignaturas, profesores, exámenes y alumnos. Como alternativa se podría utilizar una base de datos no-relacional como sería MongoDB.
- **Javascript:** Para poder realizar una aplicación que sea interactiva para el usuario, se ha escogido este lenguaje que permite la definición de eventos de esta forma podremos realizar el sistema de delimitar las preguntas a corregir. Se podría utilizar como alternativa un framework más moderno como por ejemplo, ReactJS o Angular.
- **JQuery:** Es una librería multiplataforma basado en Javascript que aunque tiene el mismo objetivo que se ha mencionado anteriormente, para ciertos casos tiene funciones que el propio Javascript no tiene y tiene una sintaxis menos verbose por lo que el código será más ligero.
- **PHP:** Nos permitirá hacer todo lo relacionado con el backend, ya sea acceder archivos del servidor o realizar todo un conjunto de acciones relacionadas con la base de datos. Como alternativa se podría utilizar Typescript ya que es un lenguaje flexible tanto en client-side (Angular) como en server-side (NodeJS).

- **HTML:** Nos permitirá estructurar y ordenar los contenidos que formarán parte de de nuestra aplicación web.
- **CSS:** Dotará de estilos a la estructura anteriormente nombrada, para poder dar una mejor experiencia de usuario.

Más relacionado con tareas de gestión del proyecto en sí, se ha decidido usar GitHub para el control de versiones y Trello para la gestión semanal de la evolución del proyecto. La selección de estas herramientas se ha basado en su gratuidad y en el uso previo de las mismas en asignaturas de la carrera.

	donde se está desarrollando la aplicación	seguir el desarrollo del proyecto en ese equipo.		
4	La aplicación no es lo suficientemente intuitiva para el usuario final.	El usuario no sabrá cómo funciona la aplicación ya que no resulta fácil de utilizar	Baja	Medio

5 REQUISITOS DEL PROYECTO

Requisitos HW

- Ordenador portátil (i5-8250U (Quad Core) / 8GB RAM / 1TB).
- Servidor local donde alojar la aplicación web y dónde instalar las librerías python necesarias.

Requisitos SW

- Sistema Operativo Windows (Linux opcional).
- Spyder (IDE para el desarrollo en Python).
- Librerías Python: openCV2, numpy, pandas, matplotlib, scikit-learn.
- PhpStorm / Sublime Text (IDE para el desarrollo de lenguajes web).
- XAMPP (Herramienta para el uso de localhost con un servidor Apache y MySQL para el desarrollo backend).

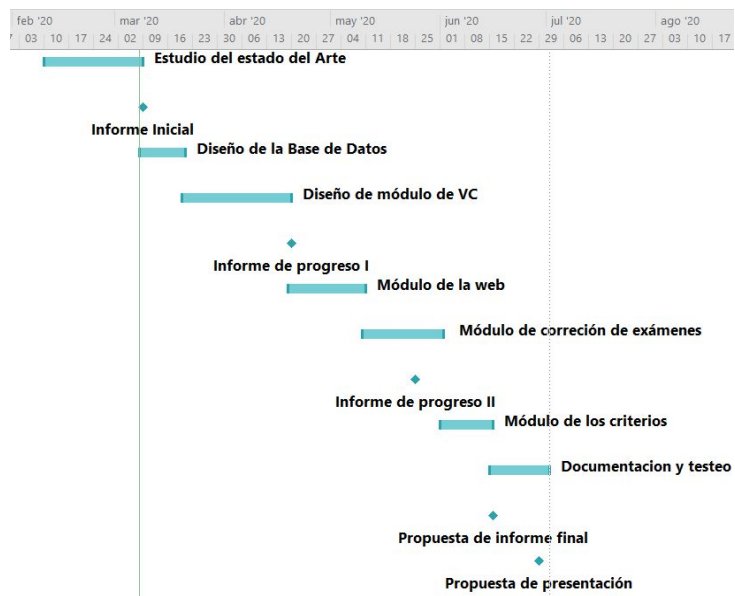
Contingencia	
ID	Descripción
1	Añadir módulo para introducir NIA alumno de manera manual
2	Añadir módulo para hacer una captura con el área designada para la pregunta en concreto.
3	Buscar un ordenador de sustitución y clonar el repositorio para poder seguir el desarrollo.
4	Utilizar iconos y diseños que sean reconocibles para los estándares de diseño de UI/UX (Véase los estándares de <i>material.io</i>)

6 RIESGOS DEL PROYECTO

Descripción			Catalogación	
ID	Descripción	Efecto	Posibilidad	Impacto
1	No funciona módulo OCR identificación alumno	No se puede saber a qué alumno corresponde cada examen	Baja	Medio
2	No se puede mostrar correctamente las preguntas que se habían delimitado	No se puede saber a qué pregunta en concreto se hacía referencia	Baja	Medio
3	Avería en el ordenador	No se puede	Baja	Bajo

7 PLANIFICACIÓN

La planificación de la ejecución de las tareas en este proyecto se han efectuado de la siguiente forma:



8 METODOLOGÍA

Para gestionar de manera correcta este proyecto se ha apostado por una metodología incremental que incluye reuniones semanales con el tutor [3]. Debido a esto, se ha optado, dentro del conjunto de metodologías Agile, por la metodología agile Kanban [4].

Esta metodología se basa en el desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Esta metodología tiene como objetivo gestionar de manera general cómo se van completando las tareas.

9 DESARROLLO

9.1. MÓDULO OCR

Como primera aproximación al desarrollo de la herramienta, se ha establecido realizar una parte de Visión por Computador mediante el uso de Deep Learning con el objetivo de realizar un módulo de OCR (Fig. 5). Dicho módulo nos permitirá transcribir los NIU que han sido escritos a mano en el examen físico, a un documento con el fin de poder hacer la correspondencia entre los NIU de los alumnos guardados en una base de datos con los que hay escritos en estos exámenes. Este proceso permitirá facilitar el tener que asignar de forma manual cada alumno con su examen. Para poder realizar estas pruebas se han utilizado un conjunto de exámenes reales. Para el entrenamiento y validación del modelo de predicción se ha decidido utilizar el dataset MNIST[5] ya que provee de muestras reales de dígitos escritos a mano, por lo que hará que la clasificación sea lo más precisa posible.

Primero se han tenido que extraer de cada examen las regiones correspondientes al NIU de forma manual (el objetivo será dotar de la opción al profesor de delimitar el área del NIU mediante la herramienta) con el objetivo de extraer la imagen lo más aislada posible para su correspondiente tratado de imagen y extracción de los dígitos que la componen (Fig. 2).

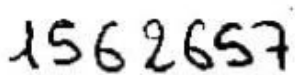


Fig. 2 Ejemplo de extracción de NIU

Para el tratado de las imágenes se ha elegido pasar estas imágenes a binario y se han invertido para así intentar asemejar lo máximo posible a lo dígitos del dataset MNIST. Para poder reducir el ruido de la imagen se ha decidido realizar una transformación morfológica, en concreto un *image opening*.

$$A \circ B = (A \ominus B) \oplus B$$

Con la finalidad de poder aislar cada uno de los dígitos de las imágenes que se han tratado, se ha utilizado la librería de Python pytesseract [6] que permite transcribir texto con fuentes existentes. Aunque no es capaz de transcribir textos a mano ya que no reconoce las letras dadas por las imágenes, se ha utilizado su funcionalidad de detección del área de los dígitos (Fig. 3).



Fig. 3 Delimitado de los dígitos de forma visual

Para poder clasificar los dígitos por separado se han redimensionado las áreas correspondientes a cada uno de los dígitos que hemos delimitado. A parte se ha guardado cada uno de los dígitos en archivos png con el objetivo de poder tratarlos más tarde.

9.1.1 DESCRIPCIÓN DE LA RED NEURONAL

Se han utilizados las librerías de Python Tensorflow [7] y Keras [8] con el objetivo de entrenar un modelo de predicción mediante una Convolutional Neural Network (CNN). Para ello se ha utilizado la librería de MNIST como conjunto de test y validation. A parte se han añadido nuestras propias imágenes con el objetivo de dotar a nuestro modelo de las características propias de estos dígitos escritos a mano.

La estructura que se ha decidido para generar el modelo de la red neuronal convolucional para esta casuística es la siguiente (Fig. 4):

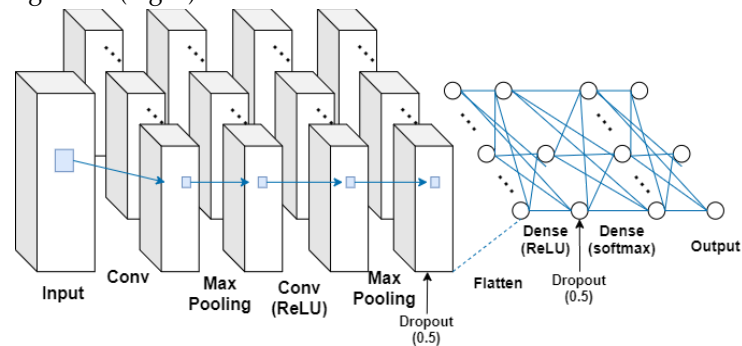


Fig. 4 Estructura de la Red Neuronal utilizada

La arquitectura de la CNN se compone principalmente de capas convolucionales seguidas de capas de max pooling y una capa fully-connected al final. Se han utilizado dropouts para evitar el overfit a la hora de entrenar. El modelo ha sido configurado con una loss function que utiliza Categorical Cross Entropy[9] y un optimizador Adam[10] para adaptar el ratio de aprendizaje.

Para el entrenamiento se utilizará la función de fit que contendrá el conjunto de train (train_X), los labels (train_Y), el conjunto de validación (test_X, test_Y), los epoch (epochs = 7) y el batch size (batch = 200). El resultado del entrenamiento con el dataset de MNIST y las imágenes de los dígitos a mano propios es el siguiente:

Accuracy: 0.9780

Loss: 0.0746

Validation Accuracy: 0.9889

Validation Loss: 0.0334

Por último, se ha cargado el modelo que previamente se ha guardado, y para cada imagen de cada dígito que compone el NIU de ese examen, se le ha etiquetado la clase a la que pertenece, en este caso el número. Estas etiquetas se han ido añadiendo a un archivo de texto que finalmente acaba conteniendo el texto correspondiente a la imagen del NIU inicial (Fig. 5), de esta forma se podrá comparar los NIU de los alumnos que hay matriculados en un asignatura con los clasificados a partir de los exámenes en físico.

1562657

Fig. 5 Resultado de la clasificación de las imágenes

Para concluir la parte del desarrollo del módulo OCR, se resume el pipeline de procesado en la Figura 6.

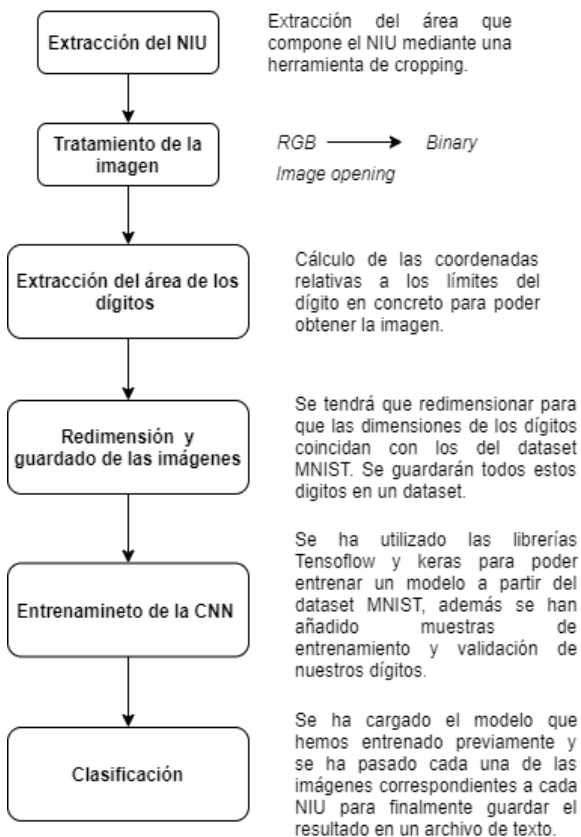


Fig. 6 Pipeline del módulo de OCR

9.2 APLICACIÓN WEB

Para la realización de la aplicación web se ha elegido utilizar MVC (Model View Controller) [11] ya que es un patrón de arquitectura del software que nos permitirá separar la lógica de negocio y de acceso a los datos almacenados en los sistemas de persistencia de la parte de la interacción entre los módulos y la parte de la interfaz gráfica y eventos (Fig. 7).

Este patrón de diseño se compone de tres componentes principales :

1. **Modelo:** Dada una llamada a través del controlador, este módulo se encargará de realizar las peticiones a los sistemas de persistencia con el objetivo de recoger los datos para poder aplicar la lógica de la aplicación. De forma generalizada este módulo se encargará de realizar las CRUD operations[12] para poder manejar los datos de la aplicación.
2. **Vista:** Este componente nos permitirá realizar la parte de interfaz de usuario y representación de los elementos visuales e incluso obtener los datos obtenidos desde el Modelo y darles sentido mediante su representación gráfica.
3. **Controlador:** Es el componente que recibe las entradas e interacciones del usuario que nos permite orquestar las llamadas a las Vistas y a parte, hacer las peticiones para recoger los datos mediante las llamadas a los Modelos.

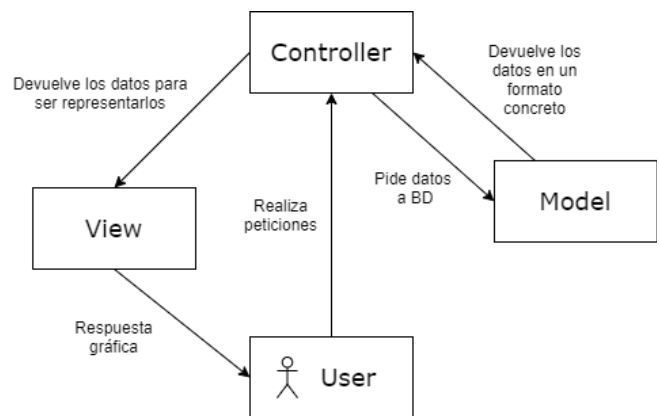


Fig. 7 Diagrama de dependencias de MVC

Se ha elegido utilizar ese patrón ya que nos permitirá encapsular nuestro código con el objetivo de evitar la repetición de código y poder desacoplar la lógica de acceso de los datos a la representación de estos, de esta forma un cambio en nuestro backend no supondrá realizar cambios en diferentes partes de nuestro código, este problema en el mundo del Diseño del Software es un Code Smell [13] llamado Shotgun Surgery [14].

9.2.1 BASE DE DATOS

El desarrollo de la aplicación web se ha basado en una base de datos SQL. Se ha decidido utilizar una base de datos relacional ya que nos permitirá crear identificadores para cada tipo de dato y poder interrelacionar los datos sin que haya duplicidad de registros. También se ha tenido en cuenta que, en caso de querer eliminar algún registro, garantizar la integridad referencial es decir, al eliminar un registro elimina todos los registros relacionados dependientes.

La base de datos que se ha utilizado para la realización de este proyecto contiene las siguientes tablas en las que se almacenarán los datos(Fig. 8):

- **Profesor** - Información de cada profesor y su identificación.
- **Alumno** - Información de cada alumno para identificarle.
- **Asignatura** - Datos del año, nombre de la asignatura, curso, semestre, números de créditos (desglosado en teoría, problemas, prácticas) y el número de grupos en los que está dividida.
- **Roles** - Información de qué rol tiene cada profesor para cada año académico.
- **Matrícula** - Relaciona cada alumno con una asignatura dentro de un año académico en concreto.
- **Prueba** - Contiene las referencias a la asignatura de donde procede y la descripción de ésta.
- **AP** - Información relacionada con la prueba hecha por un alumno y incluye la nota obtenida por el alumno en la prueba.
- **Pregunta** - Contiene las referencias de la asignatura y la prueba en la que está contenida y a parte, contiene las coordenadas que contienen la posición relativa al PDF que mostrará.
- **AP2** - Información relacionada con la pregunta respuesta por un alumno, incluye la puntuación obtenida por el alumno en la pregunta.
- **Criterio** - Información sobre un criterio de corrección aplicado a una pregunta de una prueba en concreto.
- **AC** - Cada tupla contiene información relacionada con el criterio aplicado a la respuesta de una pregunta hecha por un alumno.

Esta base de datos ha tenido que ser transformada de Oracle SQL a MySQL que, aunque mantiene gran parte de la lógica del lenguaje SQL, presenta ciertas diferencias en cuanto la sintaxis y las funciones que se utilizan. La decisión de utilizar un servidor MySQL facilita el poder conectarse a la base de datos y el envío de *queries* desde el backend de nuestra aplicación web de forma más ágil.

9.2.2 DESARROLLO DE LAS FUNCIONALIDADES

La parte más programática de la aplicación se ha comenzado con la tarea de dar forma a los datos para ser representados ya que todo el contenido que se tendrá que visualizar será referente a estos datos anteriormente descritos.

Cuando un profesor acceda a la aplicación podrá visualizar los cursos que tiene asignados (Fig. 9), a parte, podrá añadir nuevos cursos mediante un formulario que mediante un endpoint podrá insertar este curso en la base de datos.

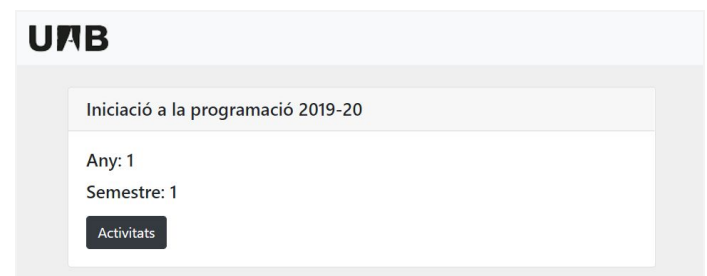


Fig. 9 Representación visual de los cursos

Si accede a uno de estos cursos, podrá visualizar todas las pruebas que están asignadas al mismo (Fig. 10). De la misma manera que se pueden crear cursos, se ha realizado un formulario con el objetivo de crear pruebas con su correspondiente documento PDF.



Fig. 8 Representación de las tablas de la Base de Datos



Fig. 10 Representación visual de las pruebas

Dentro de la pestaña de edición de la prueba se ha desarrollado un sistema interactivo que permite, mediante el cursor, crear una región rectangular que delimitaran las respuestas que se quieren visualizar en el exámen de cada alumno (Fig. 11). Esta funcionalidad se ha realizado a partir de eventos de Javascript, se han capturado las coordenadas del rectángulo a partir de un evento de *onmousemove* que guardará las coordenadas del cursor.

Al hacer click el evento *onmouseup* creará el rectángulo, que será un contenedor HTML y se hará el cálculo de las dimensiones de éste a partir de la diferencia entre las coordenadas iniciales y las actuales del cursor. Una vez creada el área, se podrá cancelar en caso de error o bien, se podrá añadir la puntuación máxima de la pregunta y guardarla en la base de datos para su posterior corrección.



Fig. 11 Delimitación del área de la pregunta mediante el cursor

Dentro de la misma sección de edición de la prueba tendremos el listado de alumnos con su información, nota de corrección hasta el momento, una opción de descargar el conjunto de notas con los nombres de los alumnos en un CSV (con el objetivo de facilitar el poder colgar las notas en el Campus Virtual / Caronte) y un botón de corregir el examen de cada alumno en particular.

Al acceder a la corrección del examen se mostrarán las preguntas que se han delimitado anteriormente mediante un elemento DOM [15] de tipo *object* que obtiene el examen del alumno a partir de una ruta creada de los datos del exámen y del propio alumno.

Este elemento DOM se ha creado mediante las dimensiones y el offset del recuadro guardado en el campo pregunta de la base de datos. De esta forma se podrá visualizar la respuesta del alumno a partir de las coordenadas asignadas al exámen vacío de una prueba en concreto (Fig. 11).

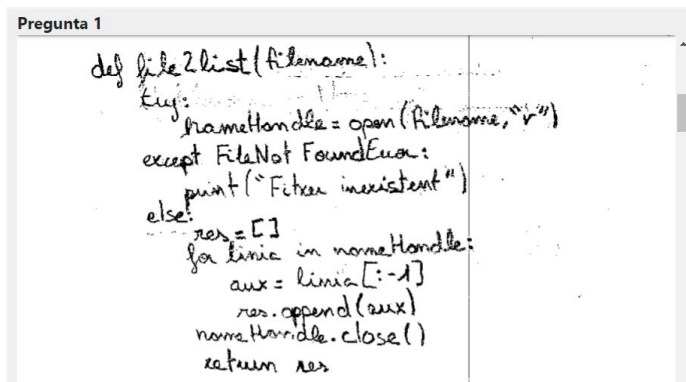


Fig. 11 Respuesta del alumno en la región asignada

En el momento que el profesor comienza la visualización de las respuestas de los alumnos podrá puntuar las mismas dentro del límite de puntuación establecido en el momento de añadir la pregunta. También tiene la opción de poder moverse sobre los exámenes mediante unas flechas de navegación que le permitirán poder avanzar o retroceder de forma que pueda elegir si quiere corregir un ejercicio a la vez para todos los alumnos sin tener que volver a la pantalla anterior para seleccionar el siguiente alumno.

Una vez puntuadas las respuestas del alumno, se podrá visualizar la nota final en el listado de alumnos para esa prueba en concreto.

10. Conclusiones

10.1 Conclusiones generales

Por parte del módulo OCR hecho mediante Deep Learning consigue reconocer parte de los NIU, pero el objetivo sería poder reconocer gran parte de los dígitos. Esto es debido a que la separación de caracteres escritos a mano varía para cada alumno, por lo que a la hora de separar los dígitos y redimensionarlos, se deforman y el sistema tiene más problemas para clasificarlos correctamente. Además, si se dotase de una base de datos de dígitos a mano de alumnos de mayor tamaño, se podría crear un modelo más ajustado a nuestras necesidades.

Por parte de la aplicación web se ha podido crear una web funcional que consigue el objetivo de agilizar la tarea de la corrección de exámenes escritos y gestionar los cursos de forma satisfactoria.

10.2 CONCLUSIONES DE PLANIFICACIÓN

La planificación de este proyecto final de carrera fué establecido en los inicios de éste. Las entregas han sido realizadas dentro del plazo salvo la parte del desarrollo de la aplicación. Esto es debido a que al ser la parte más clave de este proyecto se ha tenido que dedicar más tiempo a poder alcanzar lo puntos que la conforman. No obstante se

ha podido completar casi en su totalidad todo lo que se esperaba en los inicios del proyecto.

10.3 CONCLUSIONES DE APRENDIZAJE

Este proyecto me ha permitido poner en práctica y ampliar mis conocimientos de Deep Learning y Visión por Computador. A parte, he podido explorar el campo del reconocimiento de dígitos (OCR), algo que nunca había realizado hasta el momento y que enriquece mi entendimiento en el campo de las Ciencias de la Computación.

He conseguido ampliar mis conocimientos en las capacidades que tienen los eventos de HTML y cómo se pueden explotar a partir de Javascript.

También he comprendido en qué partes he de hacer más énfasis para poder mejorar como profesional: como el elegir tecnologías que me den más facilidades a la hora de abarcar un problema concreto e incluso tomar decisiones de diseño de software que se ajusten mejor a las necesidades que pueda tener el proyecto.

10.4 TRABAJO FUTURO Y AMPLIACIONES

Hubiese sido interesante explorar la idea de crear criterios para componer las correcciones de las preguntas individualmente, pero la incertidumbre en el planteamiento de estos y el desarrollo del core de la aplicación ha consumido gran parte del tiempo. Por este motivo este sería un claro candidato de cara a un futuro trabajo.

Al no tener una forma clara de poder consumir del módulo de identificación de NIUs de alumnos, queda desacoplado de la aplicación web por lo que una de las adiciones en un futuro sería buscar algún sistema de que la aplicación pueda consultar los resultados de la clasificación de los dígitos de los exámenes.

Siendo un proyecto que por naturaleza es bastante abierto a nuevas funcionalidades, se podrían realizar las siguientes ampliaciones:

- El uso de librerías como React[S16] podría aportar el poder separar el código en componente para su múltiple uso y de esta forma evitar la repetición de código para ciertas partes de la aplicación.
- Se podría utilizar un enfoque de DDD (Domain Driven Development) o Arquitectura Hexagonal[17], ya que permite desacoplar la lógica de negocio del modelo, de esta forma se puede focalizar en satisfacer los casos de uso sin tener que estar constantemente modificando el dominio de las entidades.
- Se podría buscar alguna alternativa para crear una API Python y de esta forma que la web pueda consumir del resultado de la clasificación de los NIU.
- También utilizar frameworks más modernos para la realización de una API REST[18] en backend que recibiría las peticiones HTTP y realizaría las consultas a la Base de Datos MySQL de forma más sencilla.

AGRADECIMIENTOS

Quiero agradecer a mi familia y amigos por darme el apoyo y darme fuerza en estos momentos tan turbulentos y tan extraños.

También quiero agradecer a mis tutores Jorge Bernal y Xavi Roca por sus consejos, feedback y su comprensión para la realización del proyecto.

BIBLIOGRAFIA

- [1] Singh, A., Karayev, S., Gutowski, K., & Abbeel, P. (2017, April). Gradescope: a fast, flexible, and fair system for scalable assessment of handwritten work. In Proceedings of the fourth (2017) acm conference on learning@ scale (pp. 81-88).
- [2] Crowdmark. 2019. Crowdmark: Online Grading Software for Instructors. Retrieved April 15, 2019 from <https://crowdmark.com/>
- [3] Manifiesto por el Desarrollo Ágil de Software. (s.f.). Recuperado 2 febrero, 2020, de <http://agilemanifiesto.org/iso/es/manifiesto.html>
- [4] Metodologías ágiles. Las 3 más usadas actualmente. (2019, 6 marzo). Recuperado 8 enero, 2020, de <https://blog.conectart.com/metodologias-agiles/>
- [5] Yann LeCun, Courant Institute, NYU. Dataset de imágenes de muestra de dígitos escritos de forma manual en formato binario. Recuperado el 22 de abril, 2020, de <http://yann.lecun.com/exdb/mnist/>
- [6] Pytesseract (Google's Tesseract-OCR Engine) Python-tesseract es una herramienta de reconocimiento óptico de caracteres de fuentes (OCR) para python. <https://pypi.org/project/pytesseract/>
- [7] TensorFlow, librería open source para la realización de proyectos y entrenamiento de modelos de Machine Learning. <https://www.tensorflow.org/>
- [8] Keras, es una API de redes neuronales de alto nivel, escrita en Python y capaz de ejecutarse sobre TensorFlow, CNTK o Theano <https://keras.io/>
- [9] Categorical Cross Entropy, es una función de pérdida que se usa en tareas de clasificación de múltiples clases. <https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions/categorical-cross-entropy>
- [10] Adam optimizer, es un algoritmo para actualizar los pesos de la red. <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>
- [11] MVC, patrón de arquitectura del software que separa la lógica vinculada a la recolección de los datos de la lógica de la representación de estos. <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>
- [12] CRUD, operaciones básicas del almacenamiento persistente para aplicaciones de bases de datos relacionales. https://en.wikipedia.org/wiki/Create,_read,_update_and_delete
- [13] Code Smells, problemas en el código en los que se puede intuir un problema más profundo o futuros problemas. <https://refactoring.guru/refactoring/smells>
- [14] Shotgun Surgery, es un Code Smell en el que el abuso de primitivas o valores no encapsulados que desencadenan realizar cambios en múltiples partes del código. <https://refactoring.guru/smells/shotgun-surgery>
- [15] DOM (Document Object Model), es una interfaz multiplataforma e independiente del lenguaje que trata un documento XML o HTML como una estructura de árbol en la que cada nodo es un objeto. https://es.wikipedia.org/wiki/Document_Object_Model
- [16] ReactJS, biblioteca de JavaScript para construir interfaces de

usuario interactivas de forma sencilla.

<https://reactjs.org/>

- [17] DDD y Arquitectura Hexagonal, enfoque para el desarrollo de software mediante una profunda conexión entre la implementación y los conceptos del modelo y lógica de negocio.
https://vaadin.com/learn/tutorials/ddd/ddd_and_hexagonal
- [18] REST, es un estilo arquitectónico de software que define un conjunto de restricciones que se utilizarán para crear servicios web.
[https://es.wikipedia.org/wiki/Transferencia de Estado Representacional](https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional)

ANEXO**PLANIFICACIÓN**

Entrega informe TFG	Fecha Inicial	Fecha Final
<i>Entrega informe inicial</i>	16/02/2020	08/03/2020
<i>Entrega informe de progreso I</i>	09/03/2020	19/04/2020
<i>Entrega informe de progreso II</i>	20/04/2020	24/05/2020
<i>Propuesta informe final</i>	25/05/2020	14/06/2020
<i>Propuesta de presentación</i>	15/06/2020	28/06/2020
<i>Entrega del dossier</i>	29/06/2020	05/07/2020
<i>Entrega poster</i>	06/07/2020	09/07/2020
<i>Defensa TFG</i>	09/07/2020	13/07/2020